

What is claimed is:


1. A method of operating a memory device driver comprising:
querying at least one memory device to discover the memory type; and
configuring the driver to access the at least one memory device according to the
discovered memory type.
2. The method of claim 1, wherein querying at least one memory device to discover
the memory type further comprises querying at least one Flash memory device to
discover the memory type.
3. The method of claim 2, wherein the at least one Flash memory device is one of a
NOR Flash and a NAND Flash.
4. The method of claim 2, wherein querying at least one memory device to discover
the memory type further comprises writing to an address of the memory device and
reading a response from the address.
5. The method of claim 2, wherein querying at least one memory device to discover
the memory type further comprises querying a common Flash interface (CFI) of the
memory device.
6. The method of claim 2, wherein querying at least one memory device to discover
the memory type further comprises querying a protection register of the memory
device.
7. The method of claim 1, wherein querying at least one memory device to discover
the memory type further comprises querying an addressable memory ID stored in
the memory device.

8. The method of claim 1, wherein querying at least one memory device to discover the memory type further comprises querying an architecture feature of the memory device.
9. The method of claim 1, wherein the driver contains at least one of a low level driver, a data manager, and a file manager.
10. The method of claim 1, wherein configuring the driver to access the at least one memory device according to the discovered memory type further comprises configuring the driver to access the at least one memory device with low level driver operation parameters and memory device command sequences to match the discovered memory type.
11. The method of claim 1, wherein configuring the driver to access the at least one memory device according to the discovered memory type further comprises configuring the driver to access the at least one memory device with data manager parameters and procedures to match the discovered memory type.
12. The method of claim 1, wherein configuring the driver to access the at least one memory device according to the discovered memory type further comprises configuring the driver to access the at least one memory device with file manager parameters and procedures to match the discovered memory type.
13. The method of claim 1, wherein configuring the driver to access the at least one memory device according to the discovered memory type further comprises configuring the driver to access the at least one memory device by matching the discovered memory type to an entry in a table and loading parameters from the table entry that matches the discovered memory type.
14. The method of claim 13, wherein the table contains at least one of a low level

driver configuration, a data manager configuration, and a file manager configuration.

15. The method of claim 13, wherein the table contains one or more entries for a selected memory type, where each of the one or more entries contains a different access data model.
16. The method of claim 13, wherein the table is modifiable to update the entries of memory types and parameters.
17. A method of operating a system comprising: /
querying at least one Flash memory device coupled to a host to discover the
memory type; and
configuring a driver routine executing on the host to access the at least one Flash
memory device according to the discovered memory type.
18. The method of claim 17, wherein the at least one Flash memory device is one of a NOR Flash and a NAND Flash.
19. The method of claim 17, wherein an interface to the Flash memory device is one of a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a multimedia card (MMC) compatible interface.
20. The method of claim 17, wherein querying at least one Flash memory device coupled to a host to discover the memory type further comprises querying a protection register of the Flash memory device.
21. The method of claim 17, querying at least one Flash memory device coupled to a host to discover the memory type further comprises querying a memory ID stored in the Flash memory device.

22. The method of claim 17, wherein querying at least one Flash memory device coupled to a host to discover the memory type further comprises querying a common Flash interface (CFI) of the Flash memory device.
23. The method of claim 17, wherein querying at least one Flash memory device coupled to a host to discover the memory type further comprises issuing a Flash command to an address and reading the result from the address.
24. The method of claim 17, wherein configuring a driver routine executing on the host to access the at least one Flash memory device according to the discovered memory type further comprises configuring the driver routine to access the at least one Flash memory device with operation parameters and memory device command sequences to match the discovered memory type.
25. The method of claim 17, configuring a driver routine executing on the host to access the at least one Flash memory device according to the discovered memory type further comprises configuring the driver routine to access the at least one Flash memory device with memory management routines to match the discovered memory type.
26. The method of claim 17, wherein configuring a driver routine executing on the host to access the at least one Flash memory device according to the discovered memory type further comprises configuring the driver routine to access the at least one Flash memory device by loading parameters from an entry in a table that matches the discovered memory type.
27. The method of claim 26, wherein the table is modifiable to update the entries of memory types and parameters.

28. The method of claim 17, wherein the host is one of a processor and an external memory controller.
29. A method of configuring a driver comprising: 
querying at least one Flash memory device to discover the memory type; and
configuring the driver to access the at least one memory device according to the
discovered memory type.
30. The method of claim 29, wherein the at least one Flash memory device is one of a
NOR Flash and a NAND Flash.
31. The method of claim 29, wherein an interface to the Flash memory device is one of
a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a
multimedia card (MMC) compatible interface.
32. The method of claim 29, wherein querying at least one Flash memory device to
discover the memory type further comprises querying a protection register of the
Flash memory device.
33. The method of claim 29, wherein querying at least one Flash memory device to
discover the memory type further comprises querying a common Flash interface
(CFI) of the Flash memory device.
34. The method of claim 29, wherein querying at least one Flash memory device to
discover the memory type further comprises writing to an address of the Flash
memory device and reading a result on the address.
35. The method of claim 29, wherein the driver contains at least one of a low level
driver, a data manager, and a file manager.

36. The method of claim 29, wherein configuring the driver to access the at least one Flash memory device according to the discovered memory type further comprises configuring the driver to access the at least one Flash memory device by matching the discovered memory type to an entry in a table and loading parameters from the table entry that matches the discovered memory type.
37. The method of claim 36, wherein the table contains at least one of a low level driver configuration, a data manager configuration, and a file manager configuration.
38. The method of claim 36, wherein the table contains one or more entries for a selected memory type, where each of the one or more entries contains a different access data model.
39. The method of claim 36, wherein an access data model for the at least one Flash memory is selected from the table.
40. A system comprising:
at least one Flash memory device; and
a host coupled to the at least one Flash memory device, wherein the host is adapted to query the at least one Flash memory device and configure a driver routine to access the at least one Flash memory device in response to the query.
41. The system of claim 40, wherein the at least one Flash memory device is one of a NAND Flash and a NOR Flash.
42. The system of claim 40, wherein an interface to the Flash memory device is one of a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a multimedia card (MMC) compatible interface.

43. The system of claim 40, wherein the host is one of a processor and an external memory controller.
44. The system of claim 40, wherein the host is adapted to query a protection register of the Flash memory device.
45. The system of claim 40, wherein the host is adapted to query a memory ID stored in the Flash memory device.
46. The system of claim 40, wherein the host is adapted to query a common Flash interface (CFI) of the Flash memory device.
47. The system of claim 40, wherein the host is adapted to query an address of the Flash memory device.
48. A machine-usable medium, the machine-usable medium containing a software routine for causing a processor to execute a method, wherein the method comprises:
querying at least one Flash memory device to discover the memory type; and
configuring a driver to access the at least one Flash memory device according to the discovered memory type.
49. The machine-usable medium of claim 48, wherein an interface to the at least one Flash memory device is one of a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a multimedia card (MMC) compatible interface.
50. The machine-usable medium of claim 48, wherein querying at least one memory device to discover the memory type further comprises querying a protection register of the Flash memory device.

51. The machine-usable medium of claim 48, wherein querying at least one memory device to discover the memory type further comprises querying a memory ID stored in the array of the Flash memory device.
52. The machine-usable medium of claim 48, wherein querying at least one memory device to discover the memory type further comprises querying a common Flash interface (CFI) of the Flash memory device.
53. A system comprising:
at least one Flash memory device; and
a host coupled to the at least one Flash memory device, wherein the host comprises
a means for detecting a Flash memory type of the at least one Flash memory device and comprises a means for configuring a driver to access the at least one Flash memory device in response to the Flash memory type detected by the means for detecting.